Equilibrium Value Method for the Proof of QIP=PSPACE

Xiaodi Wu *
Institute for Quantum Computing, University of Waterloo, Ontario, Canada,
Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, USA

March 15, 2011

Abstract

We provide an alternative proof of QIP=PSPACE to the recent breakthrough result [JJUW09]. Unlike solving some semidefinite programs that captures the computational power of quantum interactive proofs, our method starts with one QIP-Complete problem which computes the diamond norm between two admissible quantum channels. The key observation is that we can convert the computation of the diamond norm into the computation of some equilibrium value. The later problem, different from the former semidefinite programs, is of better form, easier to solve and could be interesting for its own sake. The multiplicative weight update method is also applied to solve the equilibrium value problem, however, in a relatively simpler way than the one in the original proof [JJUW09]. As a direct byproduct, we also provide a NC algorithm to compute the diamond norm of a class of quantum channels. Furthermore, we provide a generalized form of equilibrium value problems that can be solved in the same way as well as comparisons to semidefinite programs.

1 Introduction

The *interactive proof system* model, which extends the concept of *efficient proof verification*, has gradually become a fundamental notion in the theory of computational complexity since its introduction [GMR85, Bab85] in the mid 1980s. In this model, a computationally bounded *verifier* interacts with a *prover* with unbounded computational power in one or more rounds. The prover wants to convince the *verifier* to *accept*(*reject*) the input, and the verifier will make its decision based on the interacting process.

The expressive power of this kind of interactive proof system model with at most polynomial rounds of communications is characterized [She92, Sha92, LFKN92] by the well-known relationship

IP=PSPACE

through the technique commonly known as *arithmetization*. Many variants of the interactive proof system model have been studied by introducing new ingredients, such as the public-coin interactive proofs [Bab85, BM88, GS89], multi-prover interactive proofs [BOGKW88], zero-knowledge interactive proofs [GMR85, GMW91] as well as the competing-prover interactive proofs [FK97].

*The work was completed when the author was visiting the Institute for Quantum Computing , University of Waterloo as a research assistant.

This paper mainly works with the *quantum interactive proof system*, which is defined [KW00, Wat99] in a similar way to ordinary interactive proof systems except the verifier and the prover have access to quantum computers. Similar to the classical cases, several variants of quantum interactive proof systems have been studied, including the ordinary quantum interactive proofs [Wat99, KW00], public-coin quantum interactive proofs [MW05], zero-knowledge quantum interactive proofs [Wat09a, Kob08, HKSZ08], multi-prover quantum interactive proofs [KKMV09, KM03] and the competing-prover quantum interactive proofs [Gut05, GW05, GW07]. The complexity class QIP known as the problems having quantum interactive proof systems satisfies [KW00]

$$\text{PSPACE=IP} \subseteq \text{QIP} \subseteq \text{EXP}$$

Along with the introduction of the complexity class QIP, several complete problems for this complexity class have been discovered. The first complete problem, called *close images*, was first proposed in 2000 [KW00]. Several relevant problems which can be reduced to *close images* were later discovered [RW05, Ros08]. Especially, the *quantum circuits distinguishability* problem, which was proved [RW05] to be QIP-complete, serves as our start point to prove QIP=PSPACE.

Recently, a big breakthrough [JJUW09] that proves QIP=PSPACE uses the primal-dual approach [AK07] based on the *multiplicative weights update method* to solve a certain kind of semidefinite programs that characterize the computational power of QMAM. The latter complexity class was proved [MW05] to have equivalent expressive power as QIP. The multiplicative weights update method is a well-known framework (or meta-algorithm) which originates in many fields. Its matrix version, which was recently developed and discussed in a survey paper [AHK05a] and the PhD thesis of Kale [Kal07], was shown to be a great success in extending the potential applications of this famous framework. Particularly, a combinatorial primal-dual approach for solving semidefinite programs (SDP) was proposed [AK07] based on the matrix multiplicative weights update (MMW) method. Under mild conditions, the primal-dual approach can be used to improve the time performance of many known approximation algorithms via semidefinite program relaxations. The main advantage of the MMW method together with the primal-dual SDP solver for the purpose of simulating quantum complexity classes is that the resultant algorithm can be easily implemented efficiently in parallel [1]. By making use of the known result NC(poly)=PSPACE [Bor77], we can solve these SDPs in PSPACE and hence show that some quantum complexity class is contained by PSPACE. Before the proof of QIP=PSPACE [JJUW09], similar ideas were applied to show the containment QIP(2)$\subseteq$ PSPACE [JUW09] and QRG(1)$\subseteq$ PSPACE [JW09].

Unlike proving the result based on the formulation of the definition of the computational class QMAM, our proof starts with one QIP-Complete problem. The problem called *quantum circuits distinguishability* [2] computes the diamond norm between two mixed-state quantum circuits. If we were again to directly compute the diamond norm by its definition, we would encounter some SDPs or convex programs which are more complicated than the one we would have if start with QMAM. Although time-efficient algorithms have already been proposed [Wat09b, BATS09] to approximate the diamond norm, it is unknown whether these methods can also be space-efficient, namely running in PSPACE. The crucial observation here, to circumvent the problem above, is to change the form of the diamond norm before the computation. The resultant problem (see **Theorem** 1) has a very neat form and can be expressed as an equilibrium value. Most importantly,

[1] We call any algorithm efficient in parallel if it is in NC. However, in our case, the size of the matrix representation of any quantum system is exponential in the input size. Thus, NC(poly) is considered as the final complexity class.

[2] It's been observed that one can also start with another QIP-Complete problem *close images* [KW00] or the protocol to simulate QIP with competing provers in [GW05] to prove the same result in almost the same way. However, the particular choice here connects our algorithm to the computation of diamond norm which is of independent interest.

as we will see later, there is a space-efficient algorithm to solve the latter problem in PSAPCE. To our knowledge, this conversion for the first time establishes the connection between the computation of the diamond norm and the computation of some equilibrium value. Precisely, Theorem 1 claims that under certain conditions the gap between two promises of the diamond norm can be transferred to the gap between two promises of the constructed equilibrium value.

The equilibrium value, or better known as the value with *minimax-maximin* form, is an important concept in theoretical computer science. For instance, many game theory related problems can be characterized naturally in this form. The fact that we can exchange the positions of *min* and *max* in any equilibrium value makes the problem well structured and provides a simpler iterative algorithm (also based on multiplicative weight update method) to approximate the equilibrium value than the one for SDPs. Similar ideas were known in the study of game theory[3] before, and was applied [JW09] in the proof of QRG(1)$\subseteq$PSPACE. Due to the merits of the equilibrium value problem, the converted problem from the diamond norm in our paper has a relatively simpler solution (see **Theorem** 4) than the SDP considered previously [JJUW09]. As a result, our main theorem provides a simplified proof for the following fact.

Corollary 1. QIP=PSPACE

As a sequence of the connection we build between the computation of the diamond norm and the computation of some equilibrium value, we also demonstrate how our algorithm can be used to approximate the diamond norm to high precision of a class of channels efficiently in parallel. Precisely, we show

Corollary 2. *Given the classical description of any two admissible quantum channels $\mathcal{Q}_0$, $\mathcal{Q}_1$, the diamond norm of their difference $\| \mathcal{Q}_0 - \mathcal{Q}_1 \|_\diamond$ can be approximated in* NC *with inverse poly-logarithm precision*[4].

This result supplements the time-efficient algorithm for calculating diamond norm in [BATS09, Wat09b]. Although our algorithm only works for a special class of channels (also one of the most interesting cases), extensions of the current algorithm for a larger class of channels could be obtained if more complicated analysis is involved.

It is interesting to compare the proof of QIP=PSPACE in this paper and the one in [JJUW09]. Our comparison represented here is threefold. First, obviously the two approaches diverge at the start point. However, this difference is actually subtle. As we mentioned before, the QIP-Complete we considered could be replaced by the *close images* problem almost with no change of the latter proof. If one investigates the result QIP=QMAM carefully, one will find this equivalence also comes from the *close images* problem. Recall that the diamond norm problem is more naturally formulated as some SDPs or convex programs just as QMAM does. It thus seems like we *deliberately* formulate the problem by an equilibrium value instead of a more natural formulation. We consider this as the main difference between the two approaches.

Second, different formulations hence lead to the need of algorithms for different problems, SDPs and equilibrium value problems in our case. Due to the relation PSPACE=NC(poly), it suffices to find algorithms that are efficient in parallel. Fortunately, such algorithms for both problems can be obtained based on the matrix multiplicative weight update method. Nevertheless, the two algorithms are quite different in several aspects. We refer curious readers to Kale's thesis [Kal07] for complete details, while a brief comparison can be found below. We will refer the algorithm

[3] See the survey [AHK05a] for more reference.

[4] Here the input size is exactly the size of those matrices representing two channels. Thus, the precision scales down to inverse poly-logarithm.

for SDP in Kale's thesis as the primal-dual SDP solver because there are indeed other methods for solving SDP also based on the matrix multiplicative weight update method. Conceptually, the primal-dual SDP solver exploits the duality between the primal and dual problems of a certain SDP while minimax-maximin equality is made use of for equilibrium value problems. It turns out the minimax relation gives a simpler proof of the correctness of the algorithm than the duality relation. Technically, both algorithms require efficient implementation of some oracles. For those SDPs and equilibrium values about QIP, the oracle for the equilibrium value problem is easier to design than the one for SDP.[5] Furthermore, SDP solver faces an additional difficulty which is not applicable to the equilibrium value problem. As one restriction of the matrix multiplicative weight update method, any solution obtained for SDP problems only satisfies the constraints approximately. Namely, one needs to convert the raw solution into exactly satisfiable solution. However, there is no control in general about the change of the object function value after this conversion. Therefore, converting approximate solutions to exact satisfiable solutions without changing the object function value a lot is another difficulty in designing SDP solver.

Finally, it is hard to compare the performance (e.g, in terms of time, space or other resources) of those methods for general equilibrium value problems and SDPs. We do not even know to what extent those methods can be applied to general equilibrium value problems or SDPs. The analysis might heavily depend on the particular form of the problem itself. However, some progress has been made recently [Wu10, GW10] in finding efficient algorithms for a larger class of equilibrium value problems and SDPs. Particulary, there exists an equilibrium-value-based SDP solver [Wu10] in addition to the primal-dual SDP solver. The new SDP solver provides a generic way to design efficient oracles, whereas a generic way of converting approximate solution to exactly satisfiable solution remains unknown.

The rest of this paper is organized as follows. We briefly survey some preliminaries which will be useful in our proof in Section 2. The conversion from the QIP-Complete problem to some equilibrium value problem lies in Section 3, which is followed by the main proof of QIP=PSPACE in Section 4. The algorithm for computing the diamond norm is discussed in Section 5. We conclude the whole paper with the summary, Section 6, where we provide further discussions about the equilibrium value problem and some open problems. Before the readers move on to the next section, there is one point to make clear. We will not take care of the precision issues with the NC implementation in the main part of this paper. Instead, we will assume such implementation can be made exactly and deal with precision issues in Appendix B

2 Preliminaries

This section contains a summary of the fundamental notations about the useful linear-algebra facts in quantum information. For the most part of this section, it is meant to make clear the notations and the terminology used in this paper. For those readers who are not familiar with these concepts, we recommend them to refer to [Bha97, KSV02, NC00, Wat08].

A *quantum register* refers to a collection of qubits, usually represented by a complex Euclidean spaces of the form $\mathcal{X} = \mathbb{C}^{\Sigma}$ where Σ refers to some finite non-empty set of the possible states.

[5] It is not easy to compare directly since the algorithm in [JJUW09] unpacks everything and only uses one subroutine, namely *projection onto positive subspace* (the same as the one in our algorithm). Nevertheless, if one rewrites the algorithm component by component, one could find out that oracle is slightly harder to solve. Moreover, additional assumptions like the invertibility of some matrices are also necessary to solve that oracle.

For any two complex Euclidean spaces $\mathcal{X}, \mathcal{Y}$, let $\mathrm{L}(\mathcal{X}, \mathcal{Y})$ denote the space of all linear mappings (or operators) from $\mathcal{X}$ to $\mathcal{Y}$ ($\mathrm{L}(\mathcal{X})$ short for $\mathrm{L}(\mathcal{X}, \mathcal{X})$). An operator $A \in \mathrm{L}(\mathcal{X}, \mathcal{Y})$ is a *linear isometry* if $A^*A = \mathbb{1}_{\mathcal{X}}$ where A^* denotes the adjoint (or conjugate transpose) of A.

An operator $A \in \mathrm{L}(\mathcal{X})$ is *Hermitian* if $A = A^*$. The eigenvalues of a Hermitian operator are always real. For $n = \dim \mathcal{X}$, we write

$$\lambda_1(A) \geq \lambda_2(A) \geq \cdots \geq \lambda_n(A)$$

to denote the eigenvalues of A sorted from largest to smallest. An operator $P \in \mathrm{L}(\mathcal{X})$ is *positive semidefinite*, the set of which is denoted by $\mathrm{Pos}(\mathcal{X})$, if P is Hermitian and all of its eigenvalues are nonnegative, namely $\lambda_n(P) \geq 0$. An operator $\rho \in \mathrm{Pos}(\mathcal{X})$ is a *density operator*, the set of which is denoted by $\mathrm{D}(\mathcal{X})$, if it has trace equal to 1. It should be noticed that a *quantum state* of a quantum register $\mathcal{X}$ is represented by a density operator $\rho \in \mathrm{D}(\mathcal{X})$.

The Hilbert-Schmidt inner product on $\mathrm{L}(\mathcal{X})$ is defined by

$$\langle A, B \rangle = \mathrm{Tr}\, A^*B$$

for all $A, B \in \mathrm{L}(\mathcal{X})$.

A *super-operator* (or quantum channel) is a linear mapping of the form

$$\Psi : \mathrm{L}(\mathcal{X}) \to \mathrm{L}(\mathcal{Y})$$

A super-operator Ψ is said to be *positive* if $\Psi(X) \in \mathrm{Pos}(\mathcal{Y})$ for any choice of $X \in \mathrm{Pos}(\mathcal{X})$, and is *completely positive* if $\Psi \otimes \mathbb{1}_{\mathrm{L}(\mathcal{Z})}$ is positive for any choice of a complex vector space $\mathcal{Z}$. The super-operator Ψ is said to be *trace-preserving* if $\mathrm{Tr}\, \Psi(X) = \mathrm{Tr}\, X$ for all $X \in \mathrm{L}(\mathcal{X})$. A super-operator Ψ is *admissible* if it is completely positive and trace-preserving. Admissible super-operators represent the discrete-time changes in quantum systems that, in principle, can be physically realized.

One can also define the adjoint super-operator of Ψ, denoted by

$$\Psi^* : \mathrm{L}(\mathcal{Y}) \to \mathrm{L}(\mathcal{X})$$

to be the unique linear mapping that satisfies,

$$\langle B, \Psi(A) \rangle = \langle \Psi^*(B), A \rangle$$

for all operators $A \in \mathrm{L}(\mathcal{X})$ and $B \in \mathrm{L}(\mathcal{Y})$.

The *Stinespring representations* of super-operators is as follows. For any super-operator Ψ, there is some auxiliary space $\mathcal{Z}$ and $A, B \in \mathrm{L}(\mathcal{X}, \mathcal{Y} \otimes \mathcal{Z})$ such that

$$\Psi(X) = \mathrm{Tr}_{\mathcal{Z}}\, AXB^*$$

for all $X \in \mathrm{L}(\mathcal{X})$. When Ψ is admissiable, we have $A = B$ and A is a linear isometry.

A *quantum circuit* is an acyclic network of *quantum gates* connected by wires. The quantum gates represent feasible quantum operations, involving constant numbers of qubits. In a *mixed state quantum circuit* [AKN98], instead of using unitary operations as quantum gates, we allow the gates to be from any set of quantum admissible operations. In this more flexible circuit model, some part of the qubits might be discarded (or *traced out*) during the evolution of the circuit.

The *trace norm* of an operator $A \in \mathrm{L}(\mathcal{X})$ is denoted by $\|A\|_1$ and defined to be

$$\|A\|_1 = \mathrm{Tr}\, \sqrt{A^*A}$$

When A is Hermitian, we have

$$\|A\|_1 = \max\{\langle P_0 - P_1, A\rangle : P_0, P_1 \in \mathrm{Pos}(\mathcal{X}), P_0 + P_1 = \mathbb{1}_{\mathcal{X}}\}$$

The diamond norm of a super-operator $\Psi : \mathrm{L}(\mathcal{X}) \to \mathrm{L}(\mathcal{Y})$ is defined to be

$$\|\Psi\|_\diamond = \max_{\|X\|_1 \leq 1} \|\Psi \otimes \mathbb{1}_{\mathcal{X}}(X)\|_1$$

Because of including the effect of using entanglement between the input and some auxiliary space, the diamond norm serves as a good measure of the distinguishability between quantum operations. Furthermore, we can show

Fact 1. [KSV02] If a quantum channel Ψ can be represented by $\Psi(X) = \mathrm{Tr}_{\mathcal{Z}} AXB^*$ where $A, B \in \mathrm{L}(\mathcal{X}, \mathcal{Y} \otimes \mathcal{Z})$, define the new channels

$$\Psi_A(X) = \mathrm{Tr}_{\mathcal{Y}} AXA^* \quad , \quad \Psi_B(X) = \mathrm{Tr}_{\mathcal{Y}} BXB^*$$

then the diamond norm of this channel Ψ is equal to

$$\|\Psi\|_\diamond = \mathrm{F}_{\max}(\Psi_A, \Psi_B) \tag{1}$$

where

$$\mathrm{F}_{\max}(\Psi_A, \Psi_B) = \max\{\mathrm{F}(\Psi_A(\varrho), \Psi_B(\zeta)) : \varrho, \zeta \in \mathrm{D}(\mathcal{X})\}$$

and

$$\mathrm{F}(P, Q) = \|\sqrt{P}\sqrt{Q}\|_1$$

which is a generalization of the fidelity between quantum states.

3 Conversion of the QCD Problems to Equilibrium Value Problems

Definition 1 (Quantum Circuit Distinguishability)**.** For any constant a, b, such that $0 \leq b < a \leq 2$. We define a promise problem $\mathrm{QCD}_{a,b}$ as follows. Given the description of any two mixed-state quantum circuits $\mathcal{Q}_0$ and $\mathcal{Q}_1$, which are admissible quantum channels from $\mathrm{L}(\mathcal{X})$ to $\mathrm{L}(\mathcal{Y})$, exactly one of the following conditions will hold:

1. $\|\mathcal{Q}_0 - \mathcal{Q}_1\|_\diamond \geq a$
2. $\|\mathcal{Q}_0 - \mathcal{Q}_1\|_\diamond \leq b$

$\mathrm{QCD}_{a,b}$ will *accept* on the condition (1) and *reject* otherwise.

It was proved by Rosgen *et al.* [RW05] that for any constant $0 < \varepsilon < 1$, $\mathrm{QCD}_{2-\varepsilon,\varepsilon}$ is QIP-Complete. A careful reformulation of this problem will enable us to rephrase this promise problem in term of an equilibrium value problem.

Assume there exists some space $\mathcal{Z} \otimes \mathcal{Q}$ that will be constructed later, we define a min-max value $\check{\lambda}(\Xi)$ to be $\min_{\rho \in \mathrm{D}(\mathcal{X}_0 \otimes \mathcal{X}_1)} \max_{\Pi \in \Gamma} \langle \Pi, \Xi(\rho)\rangle$ where Ξ is a linear super operator mapping from $\mathrm{L}(\mathcal{X}_0 \otimes \mathcal{X}_1)$ to $\mathrm{L}(\mathcal{Z} \otimes \mathcal{Q})$ and $\Gamma = \{\Pi : 0 \leq \Pi \leq \mathbb{1}_{\mathcal{Z}\otimes\mathcal{Q}}\}$. The $\mathcal{X}_0, \mathcal{X}_1$ in the above definition are isomorphic copies of $\mathcal{X}$. Further investigation shows the value $\check{\lambda}(\Xi)$ is also an *equilibrium value*.

Given $\mathrm{D}(\mathcal{X}_0 \otimes \mathcal{X}_1)$ and Γ are convex and compact sets and the $\langle \Pi, \Xi(\rho) \rangle$ is a bilinear function over them, it follows from the well-known extensions of von' Neumann's Min-Max Theorem [vN28, Fan53] that

$$\check{\lambda}(\Xi) = \min_{\rho \in \mathrm{D}(\mathcal{X}_0 \otimes \mathcal{X}_1)} \max_{\Pi \in \Gamma} \langle \Pi, \Xi(\rho) \rangle = \max_{\Pi \in \Gamma} \min_{\rho \in \mathrm{D}(\mathcal{X}_0 \otimes \mathcal{X}_1)} \langle \Pi, \Xi(\rho) \rangle \tag{2}$$

The equilibrium value $\check{\lambda}(\Xi)$ is the quantity represented by the two sides of the above equation. Furthermore, any pair $(\check{\rho}, \check{\Pi})$ which makes the function reach the equilibrium value is called the *equilibrium point*; or, equivalently, that

$$\min_{\rho \in \mathrm{D}(\mathcal{X})} \left\langle \check{\Pi}, \Xi(\rho) \right\rangle = \left\langle \check{\Pi}, \Xi(\check{\rho}) \right\rangle = \max_{\Pi \in \Gamma} \langle \Pi, \Xi(\check{\rho}) \rangle$$

The existence of the equilibrium point follows easily from Equation [2]. Careful readers might notice the equilibrium value's form defined in this paper is related but slightly different from the one defined in the proof of QRG(1)⊂PSPACE [JW09]. In the latter's definition Γ is the set of the density operators. Thus, the equilibrium value will be the largest eigenvalue or $\mathcal{L}_\infty$ norm in some sense. However, our definition of Γ makes the equilibrium value be the summation of all positive eigenvalues. Moreover, in the situation of later discussion in this paper, the equilibrium value turns to be half the $\mathcal{L}_1$ norm.

Our main theorem of this part says the two promises of any QCD problem, or equivalently of any diamond norm of the difference of two admissible channels, can be transferred to the two promises of the value of $\check{\lambda}(\Xi)$ where Ξ can be constructed efficiently from the input to that QCD problem. Precisely,

Theorem 1. *For any instance of the* $\mathrm{QCD}_{a,b}$ *problem, there exist some space* $\mathcal{Z} \otimes \mathcal{Q}$ *and a linear super operator* Ξ *from* $\mathrm{L}(\mathcal{X}_0 \otimes \mathcal{X}_1)$ *to* $\mathrm{L}(\mathcal{Z} \otimes \mathcal{Q})$ *where the space* $\mathcal{X}_0, \mathcal{X}_1$ *are isomorphic copies of the space* $\mathcal{X}$ *such that*

$$\begin{cases} \check{\lambda}(\Xi) \leq \frac{\sqrt{4-a^2}}{2}, & \| \mathcal{Q}_0 - \mathcal{Q}_1 \|_\diamond \geq a; \\ \check{\lambda}(\Xi) \geq \frac{2-b}{2}, & \| \mathcal{Q}_0 - \mathcal{Q}_1 \|_\diamond \leq b. \end{cases}$$

where $\check{\lambda}(\Xi)$ *is the equilibrium value defined above. Further more, such a super operator* Ξ *can be constructed efficiently in parallel from the input to the* $\mathrm{QCD}_{a,b}$ *problem.*

Before we get into the proof of the theorem, it might be helpful to see where this theorem leads us to. Since for any $0 < \varepsilon < 1$ the $\mathrm{QCD}_{2-\varepsilon,\varepsilon}$ is QIP-Complete, we can choose a constant ε' such that $\check{\lambda}(\Xi)$ is either at least $\frac{2-\varepsilon'}{2}$ or at most $\frac{\sqrt{4\varepsilon'-\varepsilon'^2}}{2}$ where $\frac{2-\varepsilon'}{2} \geq \frac{\sqrt{4\varepsilon'-\varepsilon'^2}}{2}$ and there is a constant gap between the two promises. For example, if we choose $\varepsilon = 0.1$, then the two promises become

$$\text{either } \check{\lambda}(\Xi) \geq 0.95 \text{ (namely, } \|\Phi\|_\diamond \leq 0.1) \text{ or } \check{\lambda}(\Xi) \leq 0.32 \text{ (namely, } \|\Phi\|_\diamond \geq 1.9)$$

Thus, in order to simulate QIP, it suffices to compute $\check{\lambda}(\Xi)$ approximately to distinguish between the two promises. This accomplishes the conversion we need for the next step of the whole proof. A simple proof for *Theorem 1* is available below.

Proof. For any instance of $\mathrm{QCD}_{a,b}$, we are given the classical descriptions of two mixed-state quantum circuits $\mathcal{Q}_0$ and $\mathcal{Q}_1$, which are admissable quantum channels from $\mathrm{L}(\mathcal{X})$ to $\mathrm{L}(\mathcal{Y})$. Thus, we could describe the two circuits using the Stinespring representation of quantum channels. Precisely,

$$\mathcal{Q}_0(X) = \mathrm{Tr}_{\mathcal{Z}}(A_0 X A_0^*), \quad \mathcal{Q}_1(X) = \mathrm{Tr}_{\mathcal{Z}}(A_1 X A_1^*)$$

where $\mathcal{Z}$ is the auxiliary space and $A_0, A_1 \in \mathrm{L}(\mathcal{X}, \mathcal{Y} \otimes \mathcal{Z})$ are linear isometries. Then, we have,

$$\Phi(X) \stackrel{\text{def}}{=} \mathcal{Q}_0(X) - \mathcal{Q}_1(X) = \mathrm{Tr}_{\mathcal{Z}}(A_0 X A_0^* - A_1 X A_1^*) = \mathrm{Tr}_{\mathcal{Z}\otimes\mathcal{Q}}(2C_0 X C_1^*)$$

where $\mathcal{Q}$ is a complex Euclidean space of dimension 2 and $C_0, C_1 \in \mathrm{L}(\mathcal{X}, \mathcal{Y} \otimes \mathcal{Z} \otimes \mathcal{Q})$. Moreover,

$$C_0 = \frac{1}{\sqrt{2}} \begin{pmatrix} A_0 \\ A_1 \end{pmatrix}, \qquad C_1^* = \frac{1}{\sqrt{2}} \begin{pmatrix} A_0 & -A_1 \end{pmatrix}$$

It is easy to see that $C_0^* C_0 = C_1^* C_1 = \mathbb{1}_{\mathcal{X}}$ given that $A_0^* A_0 = A_1^* A_1 = \mathbb{1}_{\mathcal{X}}$. To compute the diamond norm of Φ, we define

$$\Phi_A(X) = \mathrm{Tr}_{\mathcal{Y}}(C_0 X C_0^*), \quad \Phi_B(X) = \mathrm{Tr}_{\mathcal{Y}}(C_1 X C_1^*)$$

Due to Fact 1 , we have

$$\|\Phi\|_\diamond = \mathrm{F}_{\max}(2\Phi_A, 2\Phi_B) \tag{3}$$

It is interesting and useful to see that we can use one density operator $\rho \in \mathrm{D}(\mathcal{X}_0 \otimes \mathcal{X}_1)$ to represent $\varrho, \zeta \in \mathrm{D}(\mathcal{X})$ where the space $\mathcal{X}_0, \mathcal{X}_1$ are isomorphic copies of the space $\mathcal{X}$.

Precisely, let

$$\widetilde{\Phi_A}(X) = \Phi_A(\mathrm{Tr}_{\mathcal{X}_1}(X)), \quad \widetilde{\Phi_B}(X) = \Phi_B(\mathrm{Tr}_{\mathcal{X}_0}(X))$$

It is obvious that $\widetilde{\Phi_A}, \widetilde{\Phi_B}$ are admissible quantum channels from $\mathrm{L}(\mathcal{X}_0 \otimes \mathcal{X}_1)$ to $\mathrm{L}(\mathcal{Z} \otimes \mathcal{Q})$. Define

$$\widetilde{\mathrm{F}_{\max}}(2\widetilde{\Phi_A}, 2\widetilde{\Phi_B}) = \max\{\mathrm{F}(2\widetilde{\Phi_A(\rho)}, 2\widetilde{\Phi_B(\rho)}) : \rho \in \mathrm{D}(\mathcal{X}_0 \otimes \mathcal{X}_1)\} \tag{4}$$

By taking $\rho = \varrho \otimes \zeta$ in the Equation [4], we have $\widetilde{\mathrm{F}_{\max}}(2\widetilde{\Phi_A}, 2\widetilde{\Phi_B}) \geq \mathrm{F}_{\max}(2\Phi_A, 2\Phi_B)$. To see the reverse side, we can take $\varrho = \mathrm{Tr}_{\mathcal{X}_1} \rho$ and $\zeta = \mathrm{Tr}_{\mathcal{X}_0} \rho$ in Equation [3]. Thus, we have $\widetilde{\mathrm{F}_{\max}}(2\widetilde{\Phi_A}, 2\widetilde{\Phi_B}) = \mathrm{F}_{\max}(2\Phi_A, 2\Phi_B)$. Namely,

$$\|\Phi\|_\diamond = \max\{\mathrm{F}(2\widetilde{\Phi_A(\rho)}, 2\widetilde{\Phi_B(\rho)}) : \rho \in \mathrm{D}(\mathcal{X}_0 \otimes \mathcal{X}_1)\}$$

Since in QCD problem we have the promise that either $\|\Phi\|_\diamond \geq a$ or $\|\Phi\|_\diamond \leq b$. Due to the Fuchs-van de Graaf Inequalities, for any $\varrho, \zeta \in \mathrm{D}(\mathcal{X})$,

$$1 - \frac{1}{2}\|\varrho - \zeta\|_1 \leq \mathrm{F}(\varrho, \zeta) \leq \sqrt{1 - \frac{1}{4}\|\varrho - \zeta\|_1^2} \tag{5}$$

and let $\Gamma = \{\Pi : 0 \leq \Pi \leq \mathbb{1}_{\mathcal{Z}\otimes\mathcal{Q}}\}$. By substituting $\mathrm{F}(\widetilde{\Phi_A}, \widetilde{\Phi_B})$ into Eq [5] and make use of the fact $\mathrm{F}(2\widetilde{\Phi_A}, 2\widetilde{\Phi_B}) = 2\,\mathrm{F}(\widetilde{\Phi_A}, \widetilde{\Phi_B})$, then we have when $\|\Phi\|_\diamond \geq a$,

$$\min_{\rho\in\mathrm{D}(\mathcal{X}_0\otimes\mathcal{X}_1)} \max_{\Pi\in\Gamma} \left\langle \Pi, \widetilde{\Phi_A}(\rho) - \widetilde{\Phi_B}(\rho) \right\rangle = \min_{\rho\in\mathrm{D}(\mathcal{X}_0\otimes\mathcal{X}_1)} \frac{1}{2}\|\widetilde{\Phi_A}(\rho) - \widetilde{\Phi_B}(\rho)\|_1 \leq \frac{\sqrt{4-a^2}}{2}$$

and when $\|\Phi\|_\diamond \leq b$,

$$\min_{\rho\in\mathrm{D}(\mathcal{X}_0\otimes\mathcal{X}_1)} \max_{\Pi\in\Gamma} \left\langle \Pi, \widetilde{\Phi_A}(\rho) - \widetilde{\Phi_B}(\rho) \right\rangle = \min_{\rho\in\mathrm{D}(\mathcal{X}_0\otimes\mathcal{X}_1)} \frac{1}{2}\|\widetilde{\Phi_A}(\rho) - \widetilde{\Phi_B}(\rho)\|_1 \geq \frac{2-b}{2}$$

1. Initialization: Pick a fixed $\varepsilon \leq \frac{1}{2}$, and let $W^{(1)} = \mathbb{1}_{\mathcal{X}} \in \mathrm{L}(\mathcal{X})$, $N = \dim \mathcal{X}$.
2. Repeat for each $t = 1, \ldots, T$:
   (a) Let the density operator $\rho^{(t)} = W^{(t)} / \operatorname{Tr} W^{(t)}$
   (b) Observe the loss matrix $M^{(t)} \in \mathrm{L}(\mathcal{X})$ which satisfies $-\mathbb{1}_{\mathcal{X}} \leq M^{(t)} \leq 0$ or $0 \leq M^{(t)} \leq \mathbb{1}_{\mathcal{X}}$, update the weight matrix as follows:

$$W^{(t+1)} = exp(-\varepsilon \sum_{\tau=1}^{t} M^{(\tau)})$$

Figure 1: The Matrix Multiplicative Weights Update method.

Let $\Xi = \widetilde{\Phi_A} - \widetilde{\Phi_B}$ and $\check{\lambda}(\Xi)$ be the equilibrium value defined before. Finally, we have

$$\begin{cases} \check{\lambda}(\Xi) \leq \frac{\sqrt{4-a^2}}{2}, & \| \mathcal{Q}_0 - \mathcal{Q}_1 \|_\diamond \geq a; \\ \check{\lambda}(\Xi) \geq \frac{2-b}{2}, & \| \mathcal{Q}_0 - \mathcal{Q}_1 \|_\diamond \leq b. \end{cases}$$

As we can see through the proof, the desired super operator Ξ is constructed explicitly from the input circuits $\mathcal{Q}_0, \mathcal{Q}_1$. Moreover, every step in the construction only involves fundamental operation of matrices. Due to the facts in Section 4.2, we are able to construct such Ξ efficiently in parallel. □

4 Multiplicative Weights Update method for Computing Equilibrium Values

The *multiplicative weights update method* introduced in Section 1 is a framework for algorithm design (or meta-algorithm) that works as the one shown in Fig 1. This kind of framework involves lots of technical details and we refer the curious reader to the survey and the PhD thesis mentioned in the introduction. However, for the sake of completeness, we provide the main result which will be useful in our proof. It should be noticed that $\{M^{(t)}\}$ is the freedom we have in this framework.

Theorem 2. *After T rounds, the algorithm in Fig 1 guarantees that, for any $\rho^* \in \mathrm{D}(\mathcal{X})$, we have*

$$(1-\epsilon) \sum_{\geq 0} \left\langle \rho^{(t)}, M^{(t)} \right\rangle + (1+\epsilon) \sum_{\leq 0} \left\langle \rho^{(t)}, M^{(t)} \right\rangle \leq \left\langle \rho^*, \sum_{t=1}^{T} M^{(t)} \right\rangle + \frac{lnN}{\epsilon} \tag{6}$$

Here, the subscripts ≥ 0 or ≤ 0 in the summation refer to the rounds t where $0 \leq M^{(t)} \leq \mathbb{1}$ or $-\mathbb{1} \leq M^{(t)} \leq 0$ respectively.

Since in our consideration, it always holds that $0 \leq M^{(t)} \leq \mathbb{1}$. It suffices for our purpose to prove a simpler version of the theorem 2 although the proof is almost the same as the one for the general version.

1. Let $\varepsilon = \frac{\delta}{4}$ and $T = \left\lceil \frac{16 \ln N}{\delta^2} \right\rceil$. Also let $W^{(1)} = \mathbb{1}_{\mathcal{X}}$, $N = \dim(\mathcal{X})$.
2. Repeat for each $t = 1, \ldots, T$:
   (a) Let $\rho^{(t)} = W^{(t)} / \operatorname{Tr} W^{(t)}$ and compute the $\Xi(\rho^{(t)})$. Then let $\Pi^{(t)}$ be the projection onto the positive eigenspaces of $\Xi(\rho^{(t)})$.
   (b) Let $M^{(t)} = (\Xi^*(\Pi^{(t)}) + \mathbb{1}_{\mathcal{X}})/2$, and update the weight matrix as follows:

$$W^{(t+1)} = exp(-\varepsilon \sum_{\tau=1}^{t} M^{(\tau)})$$

3. Return $\frac{1}{T} \sum_{t=1}^{T} \left\langle \Pi^{(t)}, \Xi(\rho^{(t)}) \right\rangle$ as the approximation of $\check{\lambda}(\Xi)$.

Figure 2: An algorithm that computes the approximation $\check{\lambda}(\Xi)$ with precision δ.

Theorem 3. *Assume $0 \leq M^{(t)} \leq \mathbb{1}$ for all t, after T rounds, the algorithm in Fig 1 guarantees that, for any $\rho^* \in \mathrm{D}(\mathcal{X})$, we have*

$$(1-\epsilon) \sum_{t=1}^{T} \left\langle \rho^{(t)}, M^{(t)} \right\rangle \leq \left\langle \rho^*, \sum_{t=1}^{T} M^{(t)} \right\rangle + \frac{lnN}{\epsilon} \tag{7}$$

We put off the proof in the appendix part. It will be sufficient to just remember this theorem in the first reading and skip the details.

4.1 Algorithm for Computing Equilibrium Values

Using the multiplicative weight update method to compute some kind of equilibrium values was known before, for instance the equilibrium value of zero-sum game (an algorithm to compute this value can be found in Kale's thesis [Kal07] and the survey [AHK05b]. The reference for similar algorithms with different purposes can be found in the survey [AHK05b]). However, to compute the equilibrium value defined in our form, we need to adapt the old idea to the new situation.

In order to compute the equilibrium value $\check{\lambda}(\Xi)$, we design an algorithm as shown in Fig 2. This algorithm takes the descriptions of the two mixed-state quantum circuits as input, and then compute the $\Xi = \widetilde{\Phi_A} - \widetilde{\Phi_B}$ in *Theorem 1*, and output the equilibrium value $\check{\lambda}(\Xi)$ with precision δ. Namely, the return value λ satisfies $|\lambda - \check{\lambda}(\Xi)| \leq \delta$.

Before proving the correctness of the algorithm, one might want to compare the algorithms in both Fig 1 and Fig 2. The only change in our algorithm is that we propose a way of computing $M^{(t)}$ for each round t. As we mentioned before, $\{M^{(t)}\}$ is the freedom we have in this framework. Different designs of $\{M^{(t)}\}$ can lead to different applications of this framework. For instance, the primal-dual approach for semidefinite programs in Kale's thesis [Kal07] is an example of the design of $\{M^{(t)}\}$ that provides a good application.

Theorem 4. *Using $T = \left\lceil \frac{16 \ln N}{\delta^2} \right\rceil$ rounds, the algorithm in Fig 2 returns the approximated value of $\check{\lambda}(\Xi)$ with precision δ. Namely, we have the return value λ satisfying*

$$|\lambda - \check{\lambda}(\Xi)| \leq \delta$$

Proof. First note that for any $\Pi^{(t)}$ computed during the process,

$$\forall \rho \in \mathrm{D}(\mathcal{X}), \quad |\left\langle \rho, \Xi^*(\Pi^{(t)}) \right\rangle| \leq 1$$

since $\Xi(\rho)$ is the difference between two density operators. Thus, $M^{(t)} = (\Xi^*(\Pi^{(t)}) + \mathbb{1}_{\mathcal{X}})/2$ satisfies $0 \leq M^{(t)} \leq \mathbb{1}_{\mathcal{X}}$.

Then apply *Theorem 3*, we have,

$$(1-\varepsilon) \sum_{\tau=1}^{T} \left\langle \rho^{(\tau)}, M^{(\tau)} \right\rangle \leq \left\langle \rho^*, \sum_{\tau=1}^{T} M^{(\tau)} \right\rangle + \frac{\ln N}{\varepsilon} \tag{8}$$

for any density operator $\rho^* \in \mathrm{D}(\mathcal{X})$. Substitute $M^{(t)} = (\Xi^*(\Pi^{(t)}) + \mathbb{1}_{\mathcal{X}})/2$ into Eq [8] and divide both side by T, note that $\left\langle \rho^{(t)}, M^{(t)} \right\rangle \leq 1$, then we have

$$\frac{1}{T} \sum_{\tau=1}^{T} \left\langle \rho^{(\tau)}, \Xi^*(\Pi^{(\tau)}) \right\rangle \leq \frac{1}{T} \left\langle \rho^*, \sum_{\tau=1}^{T} \Xi^*(\Pi^{(\tau)}) \right\rangle + 2\varepsilon + \frac{2\ln N}{\varepsilon T}$$

By choosing $\varepsilon = \frac{\delta}{4}$ and $T = \left\lceil \frac{16 \ln N}{\delta^2} \right\rceil$, we have

$$\lambda = \frac{1}{T} \sum_{\tau=1}^{T} \left\langle \rho^{(\tau)}, \Xi^*(\Pi^{(\tau)}) \right\rangle \leq \frac{1}{T} \left\langle \rho^*, \sum_{\tau=1}^{T} \Xi^*(\Pi^{(\tau)}) \right\rangle + \delta \tag{9}$$

In each step, $\Pi^{(t)}$ is returned as the solution to maximize $\left\langle \Pi^{(t)}, \Xi(\rho^{(t)}) \right\rangle$ for any fixed $\rho^{(t)}$. Due to the definition of the equilibrium value in Eq [2], the equilibrium value $\check{\lambda}(\Xi) \leq \left\langle \Pi^{(t)}, \Xi(\rho^{(t)}) \right\rangle$ for any returned $\Pi^{(t)}$. On the other side, choose $(\check{\rho}, \check{\Pi})$ to be any *equilibrium point* and let $\rho^* = \check{\rho}$, then we have $\left\langle \check{\rho}, \Xi^*(\Pi^{(t)}) \right\rangle \leq \check{\lambda}(\Xi)$. Using inequality [9], we have

$$\check{\lambda}(\Xi) \leq \lambda \leq \frac{1}{T} \left\langle \check{\rho}, \sum_{\tau=1}^{T} \Xi^*(\Pi^{(\tau)}) \right\rangle + \delta \leq \check{\lambda}(\Xi) + \delta \tag{10}$$

which completes the proof □

To distinguish between the two promises in *Theorem 1*, we let $\delta = 0.2$ and make use of the approximated equilibrium value returned in the algorithm. If the value is closer to 0.95, then it is the case that $\|\Phi\|_\diamond \leq 0.1$. Otherwise, it is the case that $\|\Phi\|_\diamond \geq 1.9$. Thus, we solve the promise QCD problem in this way.

4.2 Simulation by bounded-depth Boolean circuits

We denote by NC the class of promise problems computed by the logarithmic-space uniform Boolean circuits with poly-logarithmic depth. Furthermore, we denote by NC(poly) the class of promise problems computed by the polynomial-space uniform Boolean circuits with polynomial depth. Since it holds that NC(poly)=PSPACE, thus in order to simulate the algorithm above in PSPACE, it suffices to prove that we can simulate the algorithm in NC(poly).

There are a few facts about these classes which are useful in our discussion. The first fact is the functions in these classes compose nicely. It is clear that if $f \in$ NC(poly) and $g \in$ NC, then their composition $g \circ f$ is in NC(poly), which follows from the most obvious way of composing the families of circuits. Another useful fact is that many computations involving matrices can be performed by NC algorithms (Please refer to the survey [Gat93] which describes NC algorithms for these tasks). Especially, we will make use of the fact that matrix exponentials and positive eigenspace projections can be approximated to high precision in NC. A more careful treatment on those issues can be found in Appendix B.

Since we are able to perform matrix operations with sufficient accuracy in NC, it remains to show the ability to compose all the operations in NC(poly) and thus in PSPACE. Precisely,

Theorem 5. *The algorithm shown above can solve QCD problems in* NC(poly)*, and thus in* PSPACE*.*

Proof. To simulate the algorithm, it suffices to compose the following families of Boolean circuits.

1. A family of Boolean circuits that output the representation of the quantum channel Ξ (in Theorem 1) generated from the input x, namely, the descriptions of two mixed quantum circuits.

2. Follow the algorithm in Figure 2. Compose all the operations in each iteration. Consider the fact that fundamental matrix operations can be done in NC and the number of iterations $T = \left\lceil \frac{16 \ln N}{\delta^2} \right\rceil$ is polynomial in the size of x since δ is a constant and N is exponential in the size of x.

3. The circuits to distinguish between the two promises by making use of the value returned in the circuits above.

The first family is easily done in NC(poly), by computing the product of a polynomial number of exponential-size matrices which corresponds to the mixed quantum circuits. The second family is in NC by composing polynomial number of NC circuits. The third one is obviously in NC. The whole process is in NC(poly) by composing the NC(poly) and NC circuits above, and thus in PSPACE. □

It follows from the *Theorem 5* that we can solve $QCD_{1.9,0.1}$ problems in PSPACE. Since $QCD_{1.9,0.1}$ is QIP-Complete problem, and any polynomial reduction to that problem can be easily done in NC(poly) by computing the product of a polynomial number of exponential-size matrices, thus we have QIP$\subseteq$PSPACE. Combining with the known result PSPACE=IP$\subseteq$QIP, we have,

Corollary 1. QIP=PSPACE

We notice that all the proof above so far is based on the assumption that all the matrix operations can be simulated exactly. However in practice, we will need to truncate the precision to some place for some operations to be performed. Fortunately, this won't be an essential obstacle for the implementation of the algorithm. As mentioned in the introduction, all those issues will be handled in Appendix B without any change of the main result.

5 Algorithm for computing the diamond norm

Now it is our turn to discuss the computation of the diamond norm for a special class of quantum channels. Consider the following promised version of the problem first.

Definition 2 (Promised Diamond Norm Problem)**.** Given the classical description of any two admissible quantum channels $\mathcal{Q}_0, \mathcal{Q}_1$, the promised diamond norm problem PDN$(\mathcal{Q}_0, \mathcal{Q}_1, a, b)$ is asked to distinguish between the two cases , namely whether $\| \mathcal{Q}_0 - \mathcal{Q}_1 \|_\diamond \geq a$ or $\| \mathcal{Q}_0 - \mathcal{Q}_1 \|_\diamond \leq b$ where $a - b \in \Omega(1/\text{poly-log})$.

As mentioned in the introduction part, Theorem 1 explicitly makes one conversion from diamond norms to equilibrium values that perverse the promised gap. It is easy to see that Theorem 1 works for any admissible channels $\mathcal{Q}_0, \mathcal{Q}_1$. Furthermore, this conversion can be done efficiently as long as the classical descriptions of $\mathcal{Q}_0, \mathcal{Q}_1$ are given. Thus, by combing the results in Theorem 4, one can easily solve the promised version of diamond norm problems. Note that the input size changes to be the size of the matrix representing the channel now.

Proposition 6. *There is a* NC *algorithm that solves PDN*$(\mathcal{Q}_0, \mathcal{Q}_1, a, b)$ *where* $a^2 - (4b - b^2) \in \Omega(1/\textit{poly-log})$.

Proof. This is a direct consequence when one combines the result of Theorem 1 and Theorem 4. Given the promise, by Theorem 1, one can efficiently compute the equilibrium value $\check{\lambda}(\Xi)$ whose value is either no more than $\frac{\sqrt{4-a^2}}{2}$ or no less than $\frac{2-b}{2}$. Hence if the difference $a^2 - (4b - b^2) \in \Omega(1/\text{polylog})$, one can use the algorithm in Fig 2 to calculate $\check{\lambda}(\Xi)$ efficiently in parallel to sufficient precision in order to distinguish between those two cases. The NC algorithm follows directly when one composes the circuits for each step. □

The only undesired thing of this algorithm is we can only solve the problem when the condition $a^2 - (4b - b^2) \in \Omega(1/\text{polylog})$. This constraint makes it impossible for our algorithm to work for the whole range $0 \leq b \leq a \leq 2$. Evidences (implicitly in [RW05], e.g. Theorem 4.3) also demonstrate that simply repeating the channels for many times, like considering the alternative channels $(\mathcal{Q}_0 - \mathcal{Q}_1)^{\otimes N}$ or $\mathcal{Q}_0^{\otimes N} - \mathcal{Q}_1^{\otimes N}$ for some N, doesn't work either.

Fortunately, there is one conceptually easy but technically detoured approach to amplify the gap in general. Particularly, we will make use of some known properties of the quantum interactive proof systems and abuse them for a different purpose. Intuitively, we treat any quantum interactive proof protocol (assume the input is fixed) as a promised problem where the acceptance probability is either at least the completeness probability or no more than the soundness probability. Then we will convert the promised diamond norm problem into such a promised problem of one specific quantum interactive proof protocol. The crucial observation is the parallelization, amplification lemmas together with the complete problem results about quantum interactive proof systems can be exploited to amplify arbitrary gap of any general promised diamond norm problem and convert it to a new diamond norm problem[6] that can be solved by Proposition 6.

Let us demonstrate this approach with full detail. More importantly, we will show such conversion can also be computed efficiently. The latter one is due to the fact the parallelization and amplification procedures are constructed explicitly in [KW00]. Any protocol $\mathbf{P}$[7] with completeness a and soundness b will be denoted by $\mathbf{P}(a, b)$. Recall that any such protocol $\mathbf{P}(a, b)$ is treated as a promise problem where the acceptance probability is either at least a or at most b. The promised diamond norm problem PDN$(\mathcal{Q}_0, \mathcal{Q}_1, a, b)$ can thus be converted to the following protocol directly.

Definition 3. The protocol $\mathcal{P}_\diamond[\mathcal{Q}_0, \mathcal{Q}_1]$:

[6] Actually, it suffices to convert the original problem to *close images* problem. However, for the simplicity of description, we choose *QCD* problem instead.

[7] Precisely, a single prover quantum interactive proof protocol.

1. If $a^2 - (4b - b^2) \in \Omega(1/\text{polylog})$, use the algorithm in Proposition 6 to solve it directly. Otherwise, continue to next step.
2. Convert the original problem to $\mathbf{P}_\diamond[\mathcal{Q}_0, \mathcal{Q}_1](1/2 + a/4, 1/2 + b/4)$.
3. According to the parallelization lemma and amplification lemma [KW00], convert the protocol $\mathbf{P}_\diamond[\mathcal{Q}_0, \mathcal{Q}_1](1/2 + a/4, 1/2 + b/4)$ to $\mathbf{P}'_\diamond(1, 1/2)$. According to the construction implicit in the proof of the QIP-Completeness of the problem $QCD_{1.9,0.1}$, convert the protocol $\mathbf{P}'_\diamond(1, 1/2)$ to a new promised diamond norm problem $\text{PDN}(\mathcal{Q}'_0, \mathcal{Q}'_1, 1.9, 0.1)$.
4. Use the algorithm in Proposition 6 to solve the new problem $\text{PDN}(\mathcal{Q}'_0, \mathcal{Q}'_1, 1.9, 0.1)$ and return the answer.

Figure 3: Algorithm for the $\text{PDN}(\mathcal{Q}_0, \mathcal{Q}_1, a, b)$ problem.

1. The verifier receives some quantum state ρ from the prover.
2. The verifier selects $\{0, 1\}$ uniformly and applies $\mathcal{Q}_i$ to ρ and sends the result to the prover.
3. The verifier receives some j from the prover, accepts if $i = j$ and rejects otherwise.

The protocol is almost identical to the Protocol 3.2 in [RW05]. The only difference is the verifier needs to perform arbitrary admissible quantum channel $\mathcal{Q}_i$. It is not possible in general when the verifier's power is polynomial time bounded since arbitrary $\mathcal{Q}_i$ might need huge number of gates to simulate. However, it won't be an issue for us since we treat such protocol as a promised problem without its original meaning and $\mathcal{Q}_0, \mathcal{Q}_1$'s description is already given. It follows immediately from [RW05] that the protocol $\mathbf{P}_\diamond[\mathcal{Q}_0, \mathcal{Q}_1]$ has completeness $1/2 + a/4$ and soundness $1/2 + b/4$ given the promise that either $\| \mathcal{Q}_0 - \mathcal{Q}_1 \|_\diamond \geq a$ or $\| \mathcal{Q}_0 - \mathcal{Q}_1 \|_\diamond \leq b$.

The parallelization and amplification lemmas [KW00] can then be reinterpreted as a way to convert any protocol $\mathbf{P}(c, d)$ to some protocol $\mathbf{P}'(c', d')$ with desired c' and d'. This conversion can be efficiently computed when the gap between c and d is appropriate. For the promised diamond norm problem, we starts with some protocol $\mathbf{P}_\diamond(a, b)$ where the gap between a and b is at least inverse poly-logarithm and converts it to some protocol $\mathbf{P}'_\diamond(1, 1/2)$. Moreover, such conversion can be done in NC. Secondly, because $QCD_{1.9,0.1}$ is QIP-Complete problem, we can convert the protocol $\mathbf{P}'_\diamond(1, 1/2)$ again to a new promised diamond norm problem $\text{PDN}(\mathcal{Q}'_0, \mathcal{Q}'_1, 1.9, 0.1)$ where such conversion is implicitly inside the proof of QIP-Completeness of the $QCD_{1.9,0.1}$ [RW05] and the new channels $\mathcal{Q}'_0, \mathcal{Q}'_1$ can be computed in NC as well. Finally, we will invoke the algorithm in Proposition 6 to solve the new problem. To sum up,

Proposition 7. *There is a* NC *algorithm shown in Fig. 3 that solves general* $\text{PDN}(\mathcal{Q}_0, \mathcal{Q}_1, a, b)$ *problems.*

As a standard technique, an algorithm for the promised version of problems can be used as a subroutine to solve the general problems via binary search when the range of the possible results is bounded. In our case, the diamond norm between any two admissible quantum channels is bounded between 0 and 2. Hence, by recursively calling the subroutine in Fig. 3 $O(\text{poly-log})$ times, one can compute the diamond norm with $\Omega(1/\text{poly-log})$ precision.

Corollary 2. Given the classical description of any two admissible quantum channels $\mathcal{Q}_0, \mathcal{Q}_1$, the diamond norm of their difference $\| \mathcal{Q}_0 - \mathcal{Q}_1 \|_\diamond$ can be approximated in NC with inverse poly-logarithm precision.

6 Summary

In this paper, we provide an alterative proof for QIP=PSPACE which starts from one QIP-Complete problem that computes the diamond norm between two quantum admissible channels. The key observation here is to convert the computation of the diamond norm to the computation of an equilibrium value. The later problem turns out to be a more structured problem and has a good solution in NC(poly) and thus in PSPACE. Besides reducing from the QCD problem, we could also reduce from the very first QIP-Complete problem *close images* [KW00] or the protocol to simulate QIP with competing provers in [GW05]. Both reductions will lead to the similar equilibrium values to the one in this paper. The technique of computing the equilibrium values in this paper can then be applied directly and lead to the same result.

The multiplicative weights update method in our proof to solve the equilibrium value problem can be generalized to solve a class of such equilibrium value problems. Particularly, for any density operator set $\mathrm{D}(\mathcal{X})$ and another convex compact set Γ, the following general equilibrium value problem

$$\check{\lambda}(\Phi) = \min_{\rho \in \mathrm{D}(\mathcal{X})} \max_{\sigma \in \Gamma} \langle \sigma, \Phi(\rho) \rangle = \max_{\sigma \in \Gamma} \min_{\rho \in \mathrm{D}(\mathcal{X})} \langle \Phi^*(\sigma), \rho \rangle$$

can be solved efficiently in NC by the same algorithm in our paper if a good approximation algorithm to compute $\max_{\sigma \in \Gamma} \langle \sigma, \Phi(\rho^*) \rangle$ given ρ^* is available in NC and $\max_{\rho \in \mathrm{D}(\mathcal{X})} \max_{\sigma \in \Gamma} | \langle \sigma, \Phi(\rho) \rangle |$ is bounded by some poly-logarithm function.

One big open problem is to investigate to what extent the technique in this paper can be used to solve general equilibrium value problems. As mentioned in [JJUW09], it is impossible to solve arbitrary SDPs in parallel unless NC=P. It might be the same case for the general equilibrium value problems. A recent effort [Wu10, GW10] made some progress on the general form of the equilibrium value that can be solved by similar techniques and the connection between the equilibrium value problems and semidefinite programming problems. Moreover, since the first release of this paper, some deeper knowledge of variants of quantum interactive proof system is obtained. Particularly, the main open problem in [JJUW09], namely whether QRG(2)=PSPACE, is resolved [GW10] with positive answer. The class QRG(2) contains all problems which can be recognized by two-turn (i.e, one-round) quantum refereed games. The classical analogue of this class is known to coincide with PSPACE [FK97].

Another open problem is how to extend the connection we build here between the computation of diamond norms and the computation of equilibrium values. One might hope to obtain efficient parallel algorithm for calculating diamond norms of any quantum channel.

Acknowledgments

The author is grateful to Zhengfeng Ji, Yaoyun Shi and John Watrous for helpful discussions. The author also wants to thank the hospitality and invaluable guidance of John Watrous when the author was visiting the Institute for Quantum Computing, University of Waterloo. The research was completed during this visit and was supported by the Canadian Institute for Advanced Research (CIFAR). The author also wants to thank Rahul Jain, Zhengfeng Ji and anonymous reviewers for

helpful comments on the manuscript. The author's research was also partially supported by US NSF under the grants 0347078 and 0622033.

A Proof of Theorem 3

Assume $0 \leq M^{(t)} \leq \mathbb{1}$ for all t, after T rounds, the algorithm in Fig 1 guarantees that, for any $\rho^* \in \mathrm{D}(\mathcal{X})$, we have

$$(1-\epsilon)\sum_{t=1}^{T}\left\langle \rho^{(t)}, M^{(t)}\right\rangle \leq \left\langle \rho^*, \sum_{t=1}^{T} M^{(t)}\right\rangle + \frac{lnN}{\epsilon}$$

Proof. It is easy to see that all $W^{(t)} \in \mathrm{Pos}(\mathcal{X})$. Observe that, for $t = 1, \cdots, T$,

$$\begin{aligned}
\mathrm{Tr}(W^{(t+1)}) &= \mathrm{Tr}\left[\exp\left(-\varepsilon\sum_{\tau=1}^{t} M^{(\tau)}\right)\right] \leq \mathrm{Tr}\left[\exp\left(-\varepsilon\sum_{\tau=1}^{t-1} M^{(\tau)}\right)\exp\left(-\varepsilon M^{(t)}\right)\right] \\
&= \mathrm{Tr}\left[W^{(t)}\exp\left(-\varepsilon M^{(t)}\right)\right] \leq \mathrm{Tr}\left[W^{(t)}(\mathbb{1}_{\mathcal{X}} - \varepsilon' M^{(t)})\right] \\
&= \left\langle W^{(t)}, \mathbb{1}_{\mathcal{X}} - \varepsilon' M^{(t)}\right\rangle = \mathrm{Tr}\left[W^{(t)}\right](1-\varepsilon'\left\langle \rho^{(t)}, M^{(t)}\right\rangle) \\
&\leq \mathrm{Tr}\left[W^{(t)}\right]\exp(-\varepsilon'\left\langle \rho^{(t)}, M^{(t)}\right\rangle)
\end{aligned}$$

The inequality in the first line is due to Golden-Thompson inequality [Bha97]. The second inequality is due to Lemma 8 where $\varepsilon' = 1 - e^{-\varepsilon}$. The third line is obtained by substituting $\rho^{(t)} = W^{(t)}/\mathrm{Tr}\,W^{(t)}$. The final inequality is obtained by noticing that $1 - \varepsilon' x \leq e^{-\varepsilon' x}, \forall x \in [0,1]$ and $\left\langle \rho^{(t)}, M^{(t)}\right\rangle \in [0,1]$.

If we repeat the process above, by induction as well as the fact $W^{(1)} = \mathbb{1}_{\mathcal{X}}$, we have:

$$\mathrm{Tr}\left[W^{(T+1)}\right] \leq N\exp(-\varepsilon'\sum_{\tau=1}^{T}\left\langle \rho^{(\tau)}, M^{(\tau)}\right\rangle)$$

where $N = \dim(\mathcal{X})$ as defined above. On the other hand, we have

$$\mathrm{Tr}\left[W^{(T+1)}\right] = \mathrm{Tr}\left[\exp(-\varepsilon\sum_{\tau=1}^{T} M^{(\tau)})\right] \geq \exp(-\varepsilon\lambda_N(\sum_{\tau=1}^{T} M^{(\tau)}))$$

The last inequality holds because $\mathrm{Tr}(e^A) = \sum_{i=1}^{\dim(\mathcal{A})} e^{\lambda_i(A)} \geq e^{\lambda_1(A)}$. Thus, we conclude that

$$\exp(-\varepsilon\lambda_N(\sum_{\tau=1}^{T} M^{(t)})) \leq N\exp(-\epsilon'\sum_{\tau=1}^{T}\left\langle \rho^{(t)}, M^{(t)}\right\rangle)$$

Since λ_N is the minimum eigenvalue, for any density operator $\rho^* \in \mathrm{D}(\mathcal{X})$, we have $\lambda_N(\sum_{\tau=1}^{T} M^{(t)}) \leq \left\langle \rho^*, \sum_{\tau=1}^{T} M^{(t)}\right\rangle$. Take the logarithms of the both side and simplify as well as notice the fact ε' obtained by Lemma 8 has the property that $\varepsilon' \geq \varepsilon(1-\varepsilon)$, then we have:

$$(1-\varepsilon)\sum_{\tau=1}^{T}\left\langle \rho^{(\tau)}, M^{(\tau)}\right\rangle \leq \left\langle \rho^*, \sum_{\tau=1}^{T} M^{(\tau)}\right\rangle + \frac{\ln N}{\varepsilon}$$

for any density operator $\rho^* \in \mathrm{D}(\mathcal{X})$ as required. □

For the sake of completeness, we prove the lemma we used in the proof below.

Lemma 8. *For any $0 \leq \varepsilon \leq 1/2$, let $\varepsilon' = 1 - e^{-\varepsilon}$. Then we have the following matrix inequality, for any $0 \leq M \leq \mathbb{1}$,*

$$\exp(-\varepsilon M) \leq \mathbb{1} - \varepsilon' M$$

Moreover, we have $\varepsilon' \geq \varepsilon(1-\varepsilon)$.

Proof. It is easy to verify that for any $\varepsilon \in [0, 1/2]$, we have

$$f(x) \stackrel{\text{def}}{=} \exp(-\varepsilon x) \leq g(x) \stackrel{\text{def}}{=} 1 - \varepsilon' x \qquad \text{if} \quad x \in [0,1]$$

Since $0 \leq M \leq \mathbb{1}$, let $M = UDU^\dagger$ be the diagonlization of M. Then,$f(M) - g(M) = U(f(D) - g(D))U^\dagger$. Since D is a diagonal matrix of which every diagonal entry contains one eigenvalue of M and thus is in $[0,1]$, then $f(D) - g(D)$ is a diagonal matrix with non-positive diagonal due to the inequality above. Thus, $f(M) - g(M)$ is a negative semidefinite matrix and hence $f(M) \leq g(M)$, namely, $\exp(-\varepsilon M) \leq \mathbb{1} - \varepsilon' M$. It is also easy to verify that $\varepsilon' \geq \varepsilon(1-\varepsilon)$. □

B Comments on precision issues

The analysis made in the main part of this paper has assumed that all computations performed by the algorithm are exact. However, in order to implement our algorithm, some step of the computations must be approximate. Particularly, the computation of the positive eigenspace projections and the matrix exponentials will need to be approximate. An elaborated analysis on these issues can be found in [JJUW09, JW09]. We will basically follow that type of analysis and provide a brief sketch of the analysis to the specific problem in our paper.

First, it must be made clear which part of the algorithm can be made exact and which part must be made approximate. We will use the same convention of storing complex numbers as the one in [JJUW09]. Once the input x is given and stored in memory, all elementary matrix operations (in this case: addition, multiplication, and computation of the trace or partial trace) can be implemented exactly in NC [Gat93]. However, the matrix exponentials and positive eigenspace projection cannot be exact since these operations will generate irrational numbers and the precision must be truncated at somewhere. Fortunately, Watrous *et al.* [JJUW09] provided a way to approximate those two operations to high precision in NC. Precisely,

Fact 2. Given an $n \times n$ matrix M (whose operator norm bounded by k) and a positive rational number η, the computation of $n \times n$ matrix X such that $\|exp(M) - X\| < \eta$ can be done in NC.

Fact 3. Given an $n \times n$ Hermitian matrix H and a positive rational number η, the computation of an $n \times n$ positive semidefinite matrix $\Delta \leq \mathbb{1}$ such that $\|\Delta - \Lambda\| < \eta$ for Λ being the projection operator onto the positive eigenspace of H can be done in NC.

Before we move on to the analysis of the precision issue, it helps to introduce the following convention. We will represent the actual matrices generated during the algorithm by placing a tilde over the variables that represent the idealized values. As we discussed above, there are mainly two types of operations where the accuracy will be lost. Further investigation tells us that the matrix exponentials are always necessary to the multiplicative weight update method while the positive eigenspace projections are special for our application. For the generality of the

analysis, we will first discuss what the general form of Theorem 3 is when the computation is only approximate.

Consider the scheme in Fig 1 and keep the notation convention in mind. The $\tilde{\rho}^{(t)}$ will be the actual generated density operator for round t and $W^{(t+1)} = exp(-\epsilon \sum_{\tau=1}^{t} \tilde{M}^{(t)})$. The latter one is exact simply because $W^{(t)}$ is only a notation and not stored in the memory at all. Fact 2 implies that $\|\tilde{\rho}^{(t)} - W^{(t)}/\operatorname{Tr} W^{(t)}\| < \delta_1/N$ for every t where δ_1 is some constant for our purpose. The situation for $\tilde{M}^{(t)}$ is tricky in the sense that there is no idealized value for $M^{(t)}$ to have in the general scheme. By going through the proof of Theorem 3 again, we can easily obtain the following fact.

Fact 4. If the computation can only be performed approximately, the inequality in Theorem 3 becomes

$$(1-\epsilon)\sum_{t=1}^{T}\left\langle \tilde{\rho}^{(t)}, \tilde{M}^{(t)} \right\rangle \leq \left\langle \rho^*, \sum_{t=1}^{T} \tilde{M}^{(t)} \right\rangle + \frac{lnN}{\epsilon} + \frac{1}{2}T\delta_1$$

Now consider the concrete $\tilde{M}^{(t)}$ in Fig 2. By making use of the fact above we can repeat almost all the steps in the proof of Theorem 4. The only change is to replace the Equation [9] by

$$\lambda = \frac{1}{T}\sum_{\tau=1}^{T}\left\langle \tilde{\rho}^{(\tau)}, \Xi^*(\tilde{\Pi}^{(\tau)}) \right\rangle \leq \frac{1}{T}\left\langle \rho^*, \sum_{\tau=1}^{T} \Xi^*(\tilde{\Pi}^{(\tau)}) \right\rangle + \delta + \delta_1$$

By Fact 3, we have $\|\tilde{\Pi}^{(t)} - \Pi^{(t)}\| < \delta_1/N$ where the $\Pi^{(t)}$ is the projection onto the positive eigenspace of $\Xi(\tilde{\rho}^{(t)})$. Please note that $\Pi^{(t)}$ here is not its idealized value when everything is exact but rather the exact value given the approximate $\tilde{\rho}^{(t)}$. The rest part of the proof follows similarly. Finally, we will get

$$\check{\lambda}(\Xi) - \delta_1 \leq \lambda = \frac{1}{T}\sum_{\tau=1}^{T}\left\langle \tilde{\rho}^{(\tau)}, \Xi^*(\tilde{\Pi}^{(\tau)}) \right\rangle \leq \check{\lambda}(\Xi) + \delta + \delta_1$$

Since our target is to distinguish between two promises with constant gap, we can choose δ_1 to be any sufficiently small constant. In this way, the precision issues are handled.

References

[AKN98] D. Aharonov, A. Kitaev, and N. Nisan. Quantum circuits with mixed states. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 20-30, 1998

[AHK05a] S. Arora, E. Hazan, and S. Kale. Fast algorithms for approximate semidefinite programming using the multiplicative weights update method. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 339–348, 2005.

[AHK05b] S. Arora, E. Hazan, and S. Kale. The multiplicative weights update method: a meta algorithm and applications. 2005

[AK07] S. Arora and S. Kale. A combinatorial, primal-dual approach to semidefinite programs. In *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*, pages 227–236, 2007.

[Bab85] L. Babai. Trading group theory for randomness. In *Proceedings of the 17th Annual ACM Symposium on Theory of Computing*, pages 421–429, 1985.

[BATS09] A. Ben-Aroya, A. Ta-Shma. On the complexity of approximating the diamond norm, arXiv 0902.3397, 2009.

[Bha97] R. Bhatia. *Matrix Analysis*. Springer, 1997.

[BM88] L. Babai and S. Moran. Arthur-Merlin games: a randomized proof system, and a hierarchy of complexity classes. *Journal of Computer and System Sciences*, 36(2):254–276, 1988.

[BOGKW88] M. Ben-Or, S. Goldwasser, J. Kilian, and A. Wigderson. Multi-prover interactive proofs: how to remove intractability assumptions. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, pages 113–131, 1988.

[Bor77] A. Borodin. On relating time and space to size and depth. *SIAM Journal on Computing*, 6:733–744, 1977.

[Fan53] K. Fan. Minimax theorems. *Proceedings of the National Academy of Sciences*, 39:42–47, 1953.

[FK97] U. Feige and J. Kilian. Making games short. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, pages 506–516, 1997.

[Gat93] J. von zur Gathen. Parallel linear algebra. In J. Reif, editor, *Synthesis of Parallel Algorithms*, chapter 13. Morgan Kaufmann Publishers, Inc., 1993.

[GMR85] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. In *Proceedings of the 17th Annual ACM Symposium on Theory of Computing*, pages 291–304, 1985.

[GMW91] O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the ACM*, 38(1):691–729, 1991.

[GS89] S. Goldwasser and M. Sipser. Private coins versus public coins in interactive proof systems. In S. Micali, editor, *Randomness and Computation*, volume 5 of *Advances in Computing Research*, pages 73–90. JAI Press, 1989.

[Gut05] G. Gutoski Upper bounds for quantum interactive proofs with competing provers. In *Proceedings of the 20th Annual IEEE Conference on Computational Complexity*, pages 334–343, 2005.

[GW05] G. Gutoski and J. Watrous Quantum interactive proofs with competing provers. In *Proceedings of the 22th Symposium on Theoretical Aspects of Computer Science*, volume 3404 of *Lecture Notes in Computer Science*, pages 605–616. Springer 2005.

[GW07] G. Gutoski and J. Watrous Toward a general theory of quantum games. In *Proceedings of the 39th ACM Symposium on Theory of Computing*, pages 565–574, 2007.

[GW10] G. Gutoski and X. Wu Short quantum games charaterize PSPACE. available at arXiv: 1011.2787, 2010.

[HKSZ08] S. Hallgren, A. Kolla, P. Sen, and S. Zhang. Making classical honest verifier zero knowledge protocols secure against quantum attacks. In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming*, volume 5126 of *Lecture Notes in Computer Science*, pages 592–603. Springer, 2008.

[JJUW09] R. Jain, Z. Ji, S. Upadhyay, and J. Watrous. QIP = PSPACE. In *Proceedings of the 42nd ACM Symposium on Theory of Computing*, 2010.

[JUW09] R. Jain, S. Upadhyay, and J. Watrous. Two-message quantum interactive proofs are in PSPACE. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 534–543, 2009.

[JW09] R. Jain and J. Watrous. Parallel approximation of non-interactive zero-sum quantum games. In *Proceedings of the 24th IEEE Conference on Computational Complexity*, pages 243–253, 2009.

[Kal07] S. Kale. *Efficient algorithms using the multiplicative weights update method*. PhD thesis, Princeton University, 2007.

[KKMV09] J. Kempe, H. Kobayashi, K. Matsumoto, and T. Vidick. Using entanglement in quantum multi-prover interactive proofs. *Computational Complexity*, 18(2):273–307, 2009.

[KM03] H. Kobayashi and K. Matsumoto. Quantum multi-prover interactive proof systems with limited prior entanglement. *Journal of Computer and System Sciences*, 66(3):429–450, 2003.

[Kob08] H. Kobayashi. General properties of quantum zero-knowledge proofs. In *Proceedings of the Fifth IACR Theory of Cryptography Conference*, volume 4948 of *Lecture Notes in Computer Science*, pages 107–124. Springer, 2008.

[KW00] A. Kitaev and J. Watrous. Parallelization, amplification, and exponential time simulation of quantum interactive proof system. In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, pages 608–617, 2000.

[KSV02] A. Kitaev, A. Shen, M. Vyalyi Classical and Quantum Computation American Mathematical Society, 2002

[LFKN92] C. Lund, L. Fortnow, H. Karloff, and N. Nisan. Algebraic methods for interactive proof systems. *Journal of the ACM*, 39(4):859–868, 1992.

[MW05] C. Marriott and J. Watrous. Quantum Arthur-Merlin games. *Computational Complexity*, 14(2):122–152, 2005.

[NC00] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[RW05] B. Rosgen and J. Watrous. On the hardness of distinguishing mixed-state quantum computations. In *Proceedings of the 20th Conference on Computational Complexity*, pages 344-354, 2005

[Ros08] B. Rosgen. Distinguishing Short Quantum Computations. In *Proceedings of the 25th STACS*, pages 597-608, 2008

[Sha92] A. Shamir. IP = PSPACE. *Journal of the ACM*, 39(4):869–877, 1992.

[She92] A. Shen. IP = PSPACE: simplified proof. *Journal of the ACM*, 39(4):878–880, 1992.

[vN28] J. von Neumann. Zur theorie der gesellschaftsspiele. *Mathematische Annalen*, 100(1928):295–320, 1928

[Wat99] J. Watrous. PSPACE has constant-round quantum interactive proof systems. In *Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science*, pages 112–119, 1999.

[Wat08] J. Watrous. Lecture Notes for Theory of Quantum Information Fall 2008.

[Wat09a] J. Watrous. Zero-knowledge against quantum attacks. *SIAM Journal on Computing*, 39(1):25–58, 2009.

[Wat09b] J. Watrous. Semidefinite programs for completely bounded norms. *Theory of Computing* 5: 11, 2009.

[WK06] M. Warmuth and D. Kuzmin. Online variance minimization. In *Proceedings of the 19th Annual Conference on Learning Theory*, volume 4005 of *Lecture Notes in Computer Science*, pages 514–528. Springer, 2006.

[Wu10] X. Wu. Parallized solutions to semidefinite programmings in quantum complexity theory. available at arXiv.org e-Print 1009.2211.